System Verification and Validation Plan for RoCam

Team #3, SpaceY
Zifan Si
Jianqing Liu
Mike Chen
Xiaotian Lou

October 18, 2025

Revision History

Date	Version	Notes
Date 1	1.0	Initial Draft

Contents

1 Symbols, Abbreviations, and Acronyms

symbol	description
Yolo	You only look once – an existing object detection model
CNN	Convolutional Neural Network
Jetson	abbreviatio for specifcally Jetson Nano Orin, which is the proposed board for this project
ST32	abbreviation for specifically ST32 microcontroller, which is the proposed microcontroller fo

This Verification and Validation (VnV) plan establishes the evidence that our RoCam model-rocket tracking system is both built right and the right build. Verification confirms each SRS and design requirement—camera and gimbal specs, timing/latency budgets, fault handling, and code quality—via reviews, static analysis, unit/integration tests, bench and simulation runs, with full requirement-to-test traceability. Validation demonstrates the system fulfills user and mission needs in field conditions by acquiring the rocket at launch, maintaining continuous tracking through boost/coast and occlusions, and supporting a safe, simple operator workflow. Success is judged against quantitative targets and operational criteria. Deliverables include test plans/cases, automated reports, calibration and operating procedures, field-test results, and a validation summary with stakeholder sign-off.

2 General Information

2.1 Summary

The RoCam system is a ground-based camera and gimbal assembly designed to autonomously track model rockets during launch and flight. It utilizes a high-resolution camera coupled with a motorized gimbal to maintain visual lock on the rocket, providing real-time positional data and video feed. The system includes a react frontend that utilizes computer vision algorithms to detect and follow the rocket. We also desgined a ST32 microcontroller-based gimbal controller that interfaces with the Jetson board to physically move the camera for automated tracking. The primary purpose of RoCam is to replace the traditional unreliable manual tracking methods with an automated solution to produce high-quality flight data and video footage. To better assist model rocket teams across Canada to conduct successful launches by providing an affordable and reliable system that can be easily track launching movements.

2.2 Objectives

- **Performance:** sustained ≥ 60 FPS video, $1080p$ resolution, real-time streaming.
- **Reliability & Safety:** emergency stop all movements.
- **Usability:** operator can arm, acquire, and track in ≤ 4 clicks; clear status prompts.
- **Out of scope:** operate under extreme environment. Third party usage.

2.3 Challenge Level and Extras

2.4 Relevant Documentation

Author (2019)

3 Plan

3.1 Verification and Validation Team

3.2 SRS Verification

3.3 Design Verification

3.4 Verification and Validation Plan Verification

3.5 Implementation Verification

3.6 Automated Testing and Verification Tools

3.7 Software Validation

4 System Tests

4.1 Tests for Functional Requirements

4.1.1 Area of Testing1

Title for Test

1. test-id1

   Control: Manual versus Automatic
   Initial State:
   Input:
   Output:
   Test Case Derivation:
   How test will be performed:

2. test-id2

   Control: Manual versus Automatic
   Initial State:
   Input:
   Output:
   Test Case Derivation:
   How test will be performed:

4.1.2 Area of Testing2

...

4.2 Tests for Nonfunctional Requirements

4.2.1 Area of Testing1

Title for Test

1. test-id1

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input/Condition:
   Output/Result:
   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input:
   Output:
   How test will be performed:

4.2.2 Area of Testing2

...

4.3 Traceability Between Test Cases and Requirements

5 Unit Test Description

5.1 Unit Testing Scope

5.2 Tests for Functional Requirements

5.2.1 Module 1

1. test-id1

   Type:

Initial State:

Input:

Output:

Test Case Derivation:

How test will be performed:

2. test-id2

   Type:

   Initial State:

   Input:

   Output:

   Test Case Derivation:

   How test will be performed:

3. ...

5.2.2 Module 2

...

5.3 Tests for Nonfunctional Requirements

5.3.1 Module ?

1. test-id1

   Type:

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

5.3.2 Module ?

...

5.4 Traceability Between Test Cases and Modules

References

Author Author. System requirements specification. https://github.com/..., 2019.

6 Appendix

This is where you can place additional information.

6.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

6.2 Usability Survey Questions?

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning.

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

2. What pain points did you experience during this deliverable, and how did you resolve them?

3. What knowledge and skills will the team collectively need to acquire to successfully complete the verification and validation of your project? Examples of possible knowledge and skills include dynamic testing knowledge, static testing knowledge, specific tool usage, Valgrind etc. You should look to identify at least one item for each team member.

4. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?